Loop On An Array Activity

#### **Learning Outcome Addressed**

1. Code logic to loop through items in an array

**Loop on Arrays**

### Loops on Arrays

When we have a collection of elements stored in an array and have been asked to iterate over it in order to either add or remove or change any of its elements, we use a for-loop. There are also other loop methods to work with arrays(or objects) like while and for...of.

Let's see an example where we loop over a simple array:

let candidates = ['Jenna', 'Carly', 'Sofia']  
for (let i = 0; i < candidates.length; i++) {  
  // Runs 5 times, with values of each candidate.  
  console.log(`Candidate #${i}: ${candidates[i]}`);  
}  
// resulting  
// Candidate #0: Jenna  
// Candidate #1: Carly  
// Candidate #2: Sofia

Feel free to do some research on [array methods](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array).

#### Nested arrays

Arrays can have individual elements but can also have nested arrays as elements.

For example:

let oss = [['Windows', 'MacOS'], ['Seattle', 'Cupertino']]

As you see, we have two elements in the array:

* The first one is ['Windows', 'MacOS']
* The second is ['Seattle', 'Cupertino']

If we get the length of the array:

oss.length // we get 2, not 4 as you may expected

So, how do access Seattle in this case? It looks like it's the third element here.

console.log(oss[2]) //undefined, because there is no 3rd element, only 2 elements

So, what we do is access the second element and then the first:

console.log(oss[1][0]) // Seattle

Now, that you know how to loop over arrays, you are ready to pass in the array as an argument to a function.

#### Task instructions

**Flattening arrays**

Define a function called arrayFlattener, that accepts a two dimensional array as an argument.

arrayFlattener should return a new, one-dimensional array.

Check the box when complete.

Task

Define a function arrayFlattener with a 2 dimensional array as parameter. Flatten the array to one dimension.